AMENDMENTS TO THE SPECIFICATION

Please amend the section entitled "CROSS REFERENCE TO RELATED APPLICATIONS" on p. 1 of the Specification, beginning on line 6, as follows:

This disclosure is a continuation-in-part of U.S. Patent Application Number ~~[Attorney Docket No. 005306.P045]~~ 09/967,439, entitled "Method and Apparatus For Detecting Insufficient Memory For Data Extraction Processes" filed on September 28, 2001.

Please amend the paragraph beginning on line 18, on p. 1 of the Specification as follows:

Portable computing devices (also referred to herein as handheld devices) such as personal digital assistants (PDAs) available from vendors such as Palm, Handspring, Hewlett Packard, Sony, Casio, Psion, have found increasing acceptance in the business world. Some users have a need to use their handheld devices to interact with enterprise business applications such as those offered by Siebel Systems, Inc., Oracle Corporation and others. These enterprise business applications can include large databases that a number of ~~user~~ users may access and/or update at any time.

Please amend the paragraph beginning on line 3, on p. 5 of the Specification as follows:

Embodiments of an apparatus and method for ~~detecting insufficient memory conditions during data extraction processes~~ transferring information during server synchronization with a computing device are described herein. In the following description, numerous specific details are set forth to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or with other methods, components, materials, *etc.* In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

Please amend the paragraph beginning on line 14, on p. 13 of the Specification as follows:

Computer system 204 can then transfer error information to handheld device 202, as indicated by an arrow 216 in Figure 2. In one embodiment, this error information includes: (a) information on transactions that system 100 (Figure 1) does not permit the user to make transactions on data that was also changed by ~~other~~ one or more other users; and (b) information on changes made to the transactions by the server. The user can then manually correct or dispose of these errors on handheld device 202.

Please amend the paragraph beginning on line 20, on p. 15 of the Specification as follows:

In this embodiment, some of the main functions of metadata generator/extractor 301 are to determine whether sync client 310 needs updated metadata, to extract metadata that is

stored on server 116, and to transfer the extracted metadata to handheld device 300. The metadata includes definitions for screens, views, fields, *etc.*, for the handheld application (not shown) used to access local database 308. Metadata generator/extractor 301 extracts the metadata (stored in a particular format in server 116) and forms messages or datagrams containing the metadata for transmission to sync client 310.

Please amend the paragraph beginning on line 7, on p. 17 of the Specification as follows:

In this embodiment, data extractor 305 of sync engine 116 extracts database data that is visible to handheld device 300 from main database 112 (Figure 1). The term visibility has a well-known meaning in relation to remote access of databases (see for example, U.S. Patents 6,216,135 and 6,233,617). In addition, in this embodiment, data extractor 305 can avoid extracting data according to the filter settings (described previously in conjunction with arrow 218 of Figure 2). Data extractor 305 also forms the extracted database data into a file to be downloaded to handheld device 300. In one embodiment, sync engine 116 may send the file to handheld device 300 in a series of small messages or datagrams.

Please amend the paragraph beginning on line 16, on p. 19 of the Specification as follows:

In this embodiment, sync client 501 is operatively coupled to local database 308, transaction database 405, data storer 407, and datastore 409 of handheld device 120-2. The interconnection can be implemented through connection 127 (Figure 1). In addition, sync

client 501 is operatively coupled to sync log 503, client sync manager 505 and companion local database 508. In one embodiment, companion local database 508 can store an image of local database ~~405~~ 308 of handheld device 120-2. In such an embodiment, companion device 124 can be configured to synchronize companion local database 508 with main database 112 (Figure 1). A subsequent synchronization of handheld device 120-2 will synchronize its local database 308 with the now synchronized companion local database 508.

Please amend the paragraph beginning on line 3, on p. 21 of the Specification as follows:

In a block 605, the sync client can receive the application definition version. Block 605 may be omitted if the handheld application definition is included as part of the initialization data received in block 603. In one embodiment, server 116 sends the application definition version (which may have been updated) to sync client 401. Sync client 401 may compare the newly received application definition version with the application version already stored in datastore ~~401~~ 409 (*i.e.*, the current version of the handheld application). Sync client 401 then stores the new application definition version in datastore 409.

Please amend the paragraph beginning on line 11, on p. 21 of the Specification as follows:

In a block 607, the sync client can provide transaction information to the server or companion device. In one embodiment, sync client 401 sends the transaction information

stored in transaction database 405 to server 116. For example, sync ~~manger~~ client 401 may send the transaction information for all of the recorded transactions in one block of data. Alternatively, sync client 401 may send the transaction information in several smaller blocks, waiting for an acknowledgement from server 116 before sending the next smaller block. Embodiments of this operation are described in more detail below in conjunction with Figures 7 and 8.

Please amend the paragraph beginning on line 5, on p. 22 of the Specification as follows:

In a block 611, the sync client updates its local database. In one embodiment, sync client 401 requests that server 116 initiates a data extraction operation to provide updated database data to handheld device 120-3. In this embodiment, the user may cause sync client 401 to update filters before getting the database data from server 116. Thus, server 116 will avoid downloading ~~data~~ undesired database data, which helps conserve battery power and reduce the time needed to complete the synchronization process. In one embodiment, server 116 downloads the entire data extract (*i.e.*, the data visible to the sync client and after filtering) in a single large block. In an alternative embodiment, server 116 may download a relatively small block of the data extract in response to a request by sync client 401. If handheld device 120-3 properly receives this block, sync client 401 can send a request for the next block, ~~on~~ and so on until handheld device 120-3 receives the entire extract from server 116.

Please amend the paragraph beginning on line 8, on p. 27 of the Specification as follows:

In a block 819, the URL encoded transaction string is uploaded. In one embodiment, sync client 401 places the encoded transaction string in the header of an HTTP request ~~and~~ sent to server 116. If the transaction string is too large (*e.g.*, greater than two kilobytes), the string is uploaded using more than one HTTP request. After the transaction string is uploaded, the process returns to block 807. The process repeats until all of the transaction records in transaction database 405 have been processed.

Please amend the paragraph beginning on line 5, on p. 29 of the Specification as follows:

If in block 909 there is not sufficient available memory, the process proceeds to a block 913. In block 913, sync client 401 performs an error routine to gracefully exit block 609. For example, sync client 401 can cause handheld device 120-3 to display a message to the user that there is insufficient memory available to complete the synchronize process and ~~suggesting~~ suggest that the user delete unneeded files and retry the synchronization process. In another embodiment, the error routine of block 913 can prompt the user to free memory space and once the user does so, return to block 909 instead of exiting.

Please amend the paragraph beginning on line 18, on p. 30 of the Specification as follows:

In a block 1005, sync client 401 determines whether handheld device 120-3 has sufficient memory available to store the database data to be downloaded by server 116. In one embodiment, sync ~~manager 116~~ client 401 gets the size of the data extract from server 116. As previously mentioned, server 116 accesses main database 112 (Figure 1) to create an extract of database data that is visible to handheld device 120-3 and has been filtered according to the valid filter settings uploaded from the handheld device. Server 116 provides the size of the data extract (either a full extract or a delta extract) to sync client 401, which can then determine whether there is sufficient memory available in handheld device 120-3.

Please amend the paragraph beginning on line 1, on p. 31 of the Specification as follows:

If there is sufficient memory, sync client 401, in a block 1007, pulls the data extract (either full or delta) from server 116. Sync ~~manager 410~~ client 401 can pull the data extract in a single transfer or in a series of smaller transfers. In a block 1009, sync client 401 stores the extract in local database 308.

Please amend the Abstract as follows:

A method and apparatus for transferring information in synchronizing a server and a handheld device are disclosed. The information is binary information that is then compressed using a suitable compression algorithm. The compressed binary data is then text encoded using a suitable text encoding algorithm. The text encoded information is then

encoded according to a protocol associated with the connection between the server and the handheld device. For example, the server can perform the compression and encoding operations on database data to be downloaded to the handheld device to reduce the time needed to transfer the information between the server and the handheld device.